

FPGA build process for multiple tasks and platforms

C. Serrano¹, L. R. Doolittle¹, Q. Du¹, G. Huang¹,

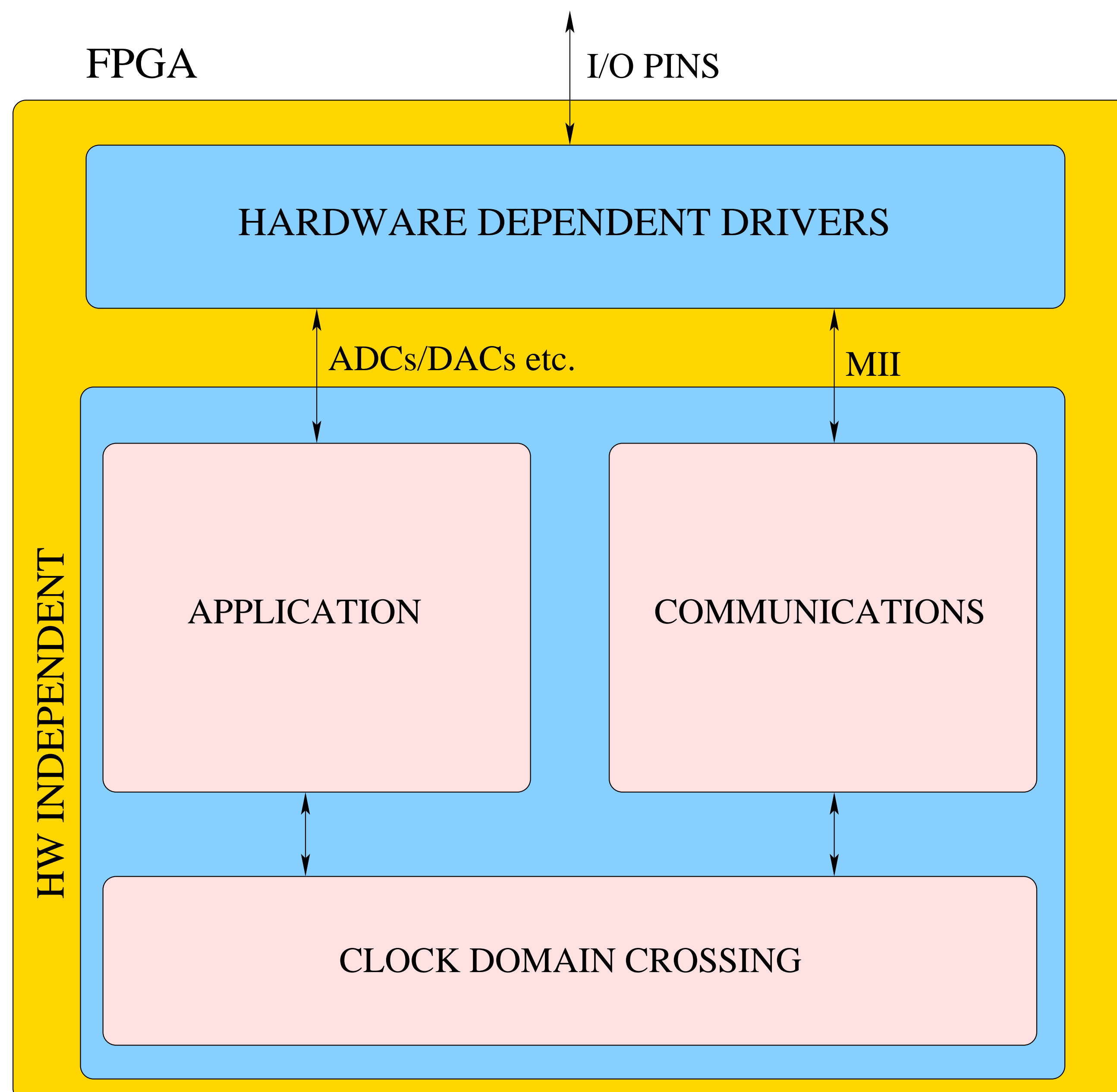
¹Lawrence Berkeley National Laboratory, Berkeley, CA 94720



Abstract

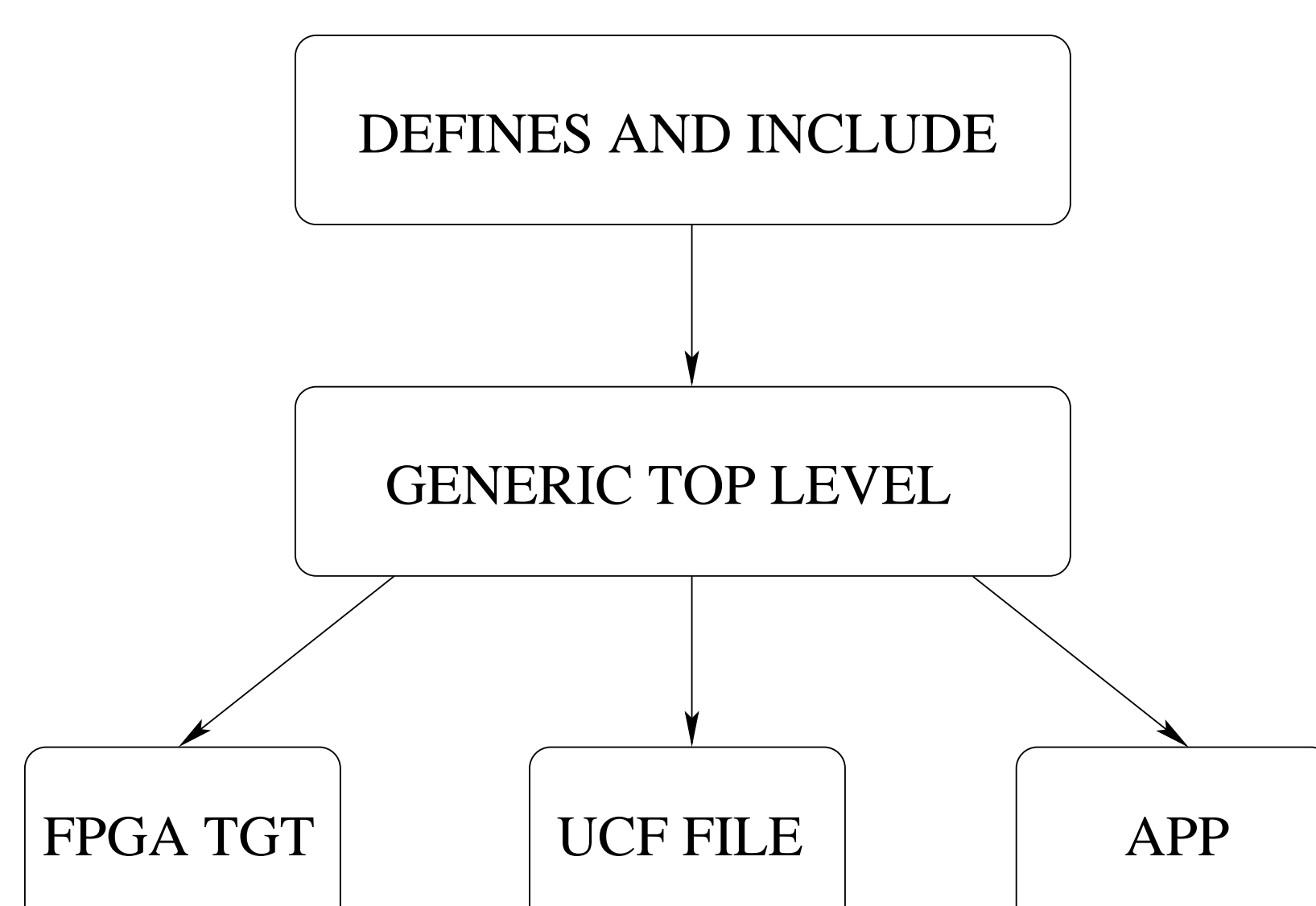
Design re-use is in general a good idea, and when it comes to developing FPGA-based systems in a variety of platforms and for different applications it becomes a necessity. LLRF and other FPGA-based systems in an accelerator normally share many common features. Signal conditioning circuits, up and down conversion, digitization, Digital Signal Processing (DSP) and digital to analog conversion are common stages in a design (each system with its particular features and implementations), along with communications with the control system. When it comes to hardware design, it is common practice to build these different stages in the signal path separately, so that one can re-use some of the common hardware components among different systems. The same applies to HDL inside an FPGA. Here we show some key concepts to bring the modularity and design re-use practice inside the FPGA, taking advantage of the flexibility of the HDL and software worlds.

Design architecture and flow



- ▶ Clean separation between:
 - ▶ Hardware-dependent logic and drivers,
 - ▶ Application and communications.
- ▶ Well-defined, standard interfaces.
- ▶ Full control over development flow:
 - ▶ Standard Unix/Linux tools: Make, awk, python, etc.,
 - ▶ Primarily open source electronic design tools (gEDA: gschem, pcb, gerbv, icarus, gtkwave),
 - ▶ Use of well-established standards (communication protocols, programming languages, tools, etc.),
 - ▶ Use of libraries and version control software,
 - ▶ HDL-Make for automatic Makefile generation,
 - ▶ Single source of truth register map configuration file shared between HDL and software automatic generation tool.
- ▶ Main blocks are inter-changeable and support is available for a large number of FPGA platforms.

Multiple applications and targets



Conclusions

- ▶ Considerable amount of FPGA code base built up over years of development.
- ▶ Work flow matches well the diversity of projects, applications and target platforms we need to support.
- ▶ Full control over development flow enabling collaboration and publishing.
- ▶ Use of Open Source tools wherever possible.
- ▶ Mixture of in-house and standard tools brings good balance.